## CLAIMS

What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A method of managing free memory, said method comprising:

creating a free list having entries with addresses of free memory locations; and

caching a portion of said free list in a cache having entries with addresses of free memory locations, wherein said cache includes a first threshold and a second threshold.

2. The method of claim 1, wherein said entries in said cache include valid entries and modified entries, and wherein said caching step further comprises:

moving a block of entries from said free list to said cache when a number of valid entries in said cache reaches said first threshold; and

moving a block of entries from said cache to said free list when the number of valid entries in said cache reaches said second threshold.

3. The method of claim 2, wherein said first threshold is set equal to the number of entries in said block of entries read from said free list to said cache less one entry.

4. The method of claim 3, wherein said first threshold is set at 7 entries.

5. The method of claim 2, wherein said second threshold is set equal to the maximum number of entries in said cache less the number of entries in said block of entries written from said cache to said free list.

6. The method of claim 5, wherein said second threshold is set at 24 entries.

7. The method of claim 1 further comprising the step of writing a magic pattern to each entry in said free list.

8. The method of claim 7, wherein said magic pattern is written to the last 8 bits of each entry in said free list.

9. The method of claim 1 further comprising the step of writing entries from said cache to a global cache.

10. The method of claim 9, wherein the act of the entries from said cache to the global cache comprises writing entries from a plurality of caches to said global cache, wherein said plurality of caches are associated with a plurality of memory banks in a plurality of memory channels.

11. The method of claim 10, wherein said entries written to said global cache are distributed between said memory banks and said memory channels.

12. A method of managing free memory in a memory system having a plurality of memory banks in a plurality of memory channels, said method comprising:

creating a plurality of free lists having entries with addresses of free memory locations, wherein a free list is created for each memory bank in each memory channel; and

writing entries from said free lists to a global cache, wherein said entries are written to said global cache are distributed between memory channels and memory banks.

13. The method of claim 12 wherein said entries written from said free lists to said global cache are written sequentially between said memory channels and said memory banks.

14. The method of claim 12 further comprising:

caching portions of said free lists in a plurality of caches before writing said entries to said global cache,

wherein each cache is associated with a memory bank in each memory channel.

15. The method of claim 14, wherein each cache includes an upper threshold and a lower threshold.

16. The method of claim 15, wherein each cache includes valid entries and modified entries, and wherein said caching step further comprises:

reading a block of entries from a free list to said cache when the number of valid entries in said cache reaches said lower threshold; and

writing a block of entries from said cache to said free list when the number of modified entries in said cache reaches said upper threshold.

17. The method of claim 16, wherein said lower threshold is set equal to the number of entries in said block of entries read from said free list to said cache less one entry, and wherein said upper threshold is set equal to the maximum number of entries in said cache less the number of entries in said block of entries written from said cache to said free list.

18. A method of managing free memory, said method comprising:

creating a plurality of free lists having entries with addresses of free memory locations, wherein said free lists are associated with a plurality of memory banks in a plurality of memory channels;

writing entries from said free lists to a plurality of caches; and

writing entries from said caches to a global cache, wherein said entries written to said global cache are distributed between said memory channels and said memory banks.

19. The method of claim 18, wherein said entries from said caches are written to said global cache sequentially between said memory channels and said memory banks.

20. A system for managing free memory comprising:

a free list having entries with addresses of free memory locations; and

a cache configured to receive a portion of said free list, wherein said cache includes a first threshold and a second threshold.

21. The system of claim 20, wherein each entry in said free list includes a magic pattern.

22. The system of claim 20 further comprising a global cache configured to receive entries from said cache.

23. The system of claim 22 further comprising a plurality of caches connected to said global cache, wherein said plurality of caches are connected to a plurality of memory banks in a plurality of memory channels.

24. The system of claim 23, wherein said global cache is configured to receive entries distributed between said plurality of caches.

25. A method of managing allocation of free memory, said method comprising:

providing a free list having a first set of addresses of free memory locations;

providing a bank cache having a second set of addresses of free memory locations;

providing a global cache having a third set of addresses of free memory locations; and

moving a plurality of entries from the free list to the bank cache if a current number of entries in the bank cache is less than a lower threshold;

wherein the first, second and third sets of addresses combine to represent the free memory.

26. The method of managing allocation of free memory of claim 25 further comprising moving a plurality of entries from the bank cache to the free list if the current number of entries in the bank cache is greater than an upper threshold.

27. The method of managing allocation of free memory of claim 25 further comprising moving an entry from the bank cache to the global cache if the global cache is not full.

28. The method of managing allocation of free memory of claim 25 further comprising removing an entry from the global cache when an entry is allocated.

29. The method of managing allocation of free memory of claim 25 further comprising adding an entry to the bank cache when the entry is de-allocated.

30. A method of managing allocation of free memory, wherein the free memory is represented by a plurality of addresses, said method comprising:

providing a plurality of memory modules each associated with a section of memory the free memory, wherein each memory module includes a free list containing a first list of entries of free memory addresses in the section of memory, and wherein each memory module further includes an associated bank cache containing a second list of entries of free memory addresses in the section of memory;

providing a global cache containing a third list of entries of free memory addresses of the free memory, wherein the third list includes entries of free memory address from a plurality of the sections of memory; and

maintaining a list of distributed entries among the memory modules by moving an entry to the global cache from a changing one of the associated bank caches if the global cache is not full;

wherein the plurality of first and second lists and the third list combine to represent the free memory.

31. The method of managing allocation of free memory of claim 30 further comprising moving a plurality of entries from one of the free list to the associated bank cache if a current number of entries in the associated bank cache is less than a first threshold.

32. The method of managing allocation of free memory of claim 30 further comprising moving a plurality of entries from one of the associated bank cache to the free list if a current number of entries in the associated bank cache is greater than a second threshold.

33. The method of managing allocation of free memory of claim 30 further comprising removing an entry from the global cache when an entry is allocated.

34. The method of managing allocation of free memory of claim 30 further comprising adding an entry to the bank cache when the entry is de-allocated.

35. A method of managing allocation of free memory, said method comprising:

providing a free list having a first set of addresses of free memory locations;

providing a bank cache having a second set of addresses of free memory locations;

providing a global cache having a third set of addresses of free memory locations;

moving a plurality of entries from the free list to the bank cache when a current number of entries in the bank cache is less than a lower threshold;

moving a plurality of entries from the bank cache to the free list when the current number of entries in the bank cache is greater than an upper threshold;

moving an entry from the bank cache to the global cache if the global cache is not full;

removing an entry from the global cache when an entry is allocated; and

adding an entry to the bank cache when the entry is de-allocated.

36. A method of managing allocation of free memory, wherein the free memory is represented by a plurality of addresses, said method comprising:

providing a plurality of memory modules each associated with a section of memory the free memory, wherein each memory module includes a free list containing a first list of entries of free memory addresses in the section of memory, and wherein each memory module further includes an associated bank cache containing a second list of entries of free memory addresses in the section of memory;

providing a global cache containing a third list of entries of free memory addresses of the free memory, wherein the third list includes entries of free memory address from a plurality of the sections of memory;

moving a plurality of entries from one of the free list to the associated bank cache if the current number of entries in the associated bank cache is less than a first threshold;

moving a plurality of entries from one of the associated bank cache to the free list if the current number of entries in the associated bank cache is greater than an second threshold;

moving an entry from the associated bank cache of a changing one of the memory modules to the global cache to create a distributed list if the global cache is not full;

removing an entry from the global cache when an entry is allocated; and

adding an entry to the bank cache when the entry is de-allocated.

37. A method of distributing free memory addresses, said method comprising:

providing a plurality of free list and bank cache pairs, wherein each pair is associated with a subsection of memory, and wherein the free list and the bank cache each contain entries, wherein each entry represents a free memory address within the subsection of memory; and

moving one or more entries in one of the pairs from the free list to the bank cache if a current number of entries in the bank cache is less than a first threshold.

38. The method of distributing free memory addresses of claim 37, wherein each of the free lists initially contains entries which represent all of free memory within the subsection of memory.

39. The method of distributing free memory addresses of claim 37, said method further comprising:

providing a global cache for containing entries representing free memory addresses; and

moving an entry from a changing one of the bank caches to the global cache.

40. The method of distributing free memory addresses of claim 39, said method further comprising removing an entry from the global cache when an entry is allocated.

41. The method of distributing free memory addresses of claim 37, said method further comprising adding an entry to the bank cache when the entry is de-allocated.

42. The method of distributing free memory addresses of claim 37, said method further comprising moving one or more entries to the free list from the bank cache if the current number of entries in the bank cache is greater than a second threshold.